SJCE_HackOverflow – Test Results

As discussed in Solution Design & Architecture, we are applying our solution to test our idea. Since the time given to us was limited & as we require huge amount of data for forecasting, we couldn't create the dataset with our hardware designed.

For illustration of our idea, we have used publicly available datasets.

i) For Daily forecasting – **(Filename-daily_data.csv)**

https://datahack.analyticsvidhya.com/contest/practice-problem-time-series-2/

ii) For Monthly forecasting – **(Filename-monthly_ preprocessed.csv)**

https://www.kaggle.com/rakannimer/air-passengers/data

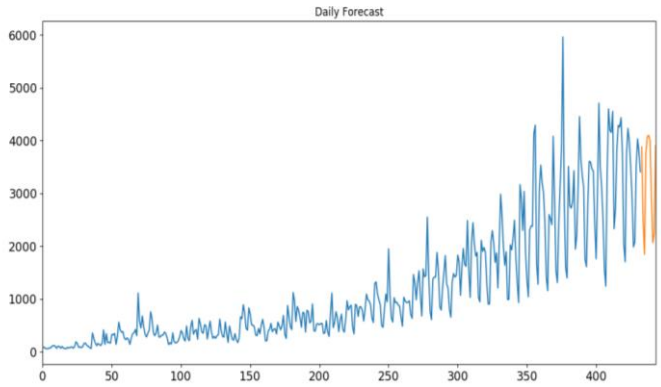
Dataset – 1 – After Resampling the daily data file. (Filename = daily_preprocessed.csv)

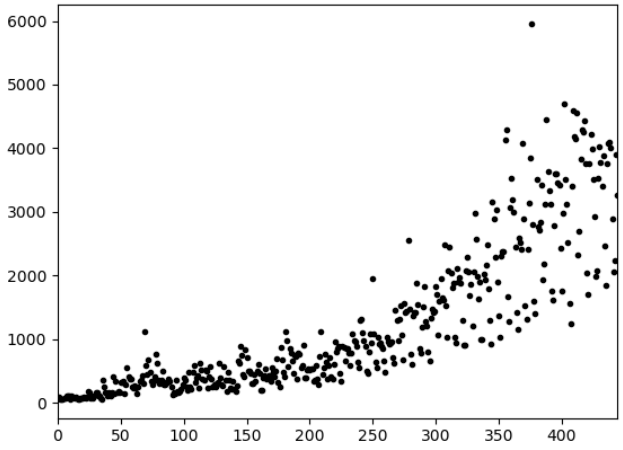
Number of training samples – **433 Days**

Number of testing samples – **10 days**

Y **– Count (No. of people entering the place).**

Data Visualization (Filename- DataVisualize.py)

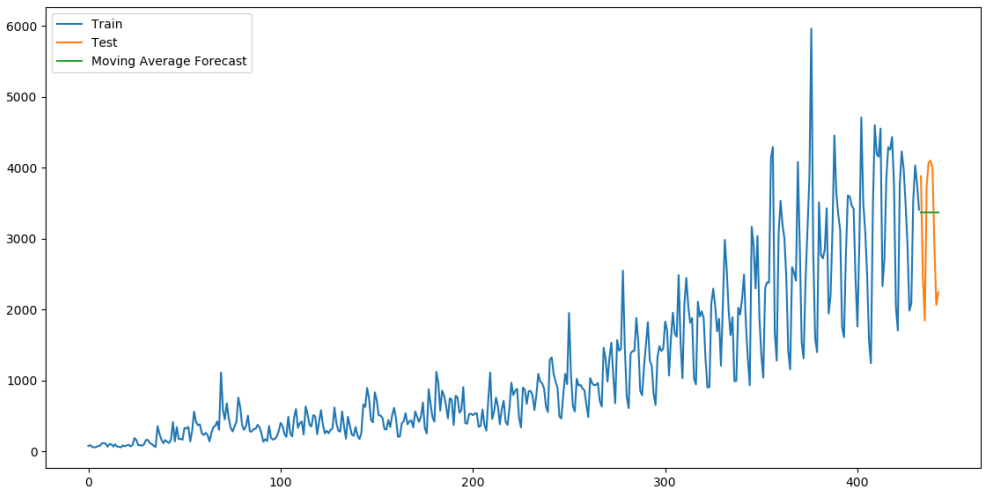




From the above plots it is clear that there is an increasing trend, so by inspection, we can tell that the Simple/ Naïve methods will not produce better results.

Model-1-Moving Average **(Filename – sma.py)**

$$\hat{y}_i = \frac{1}{p}(y_{i-1} + y_{i-2} + y_{i-3} \ldots\ldots + y_{i-p})$$



RMSE= 902.052770075

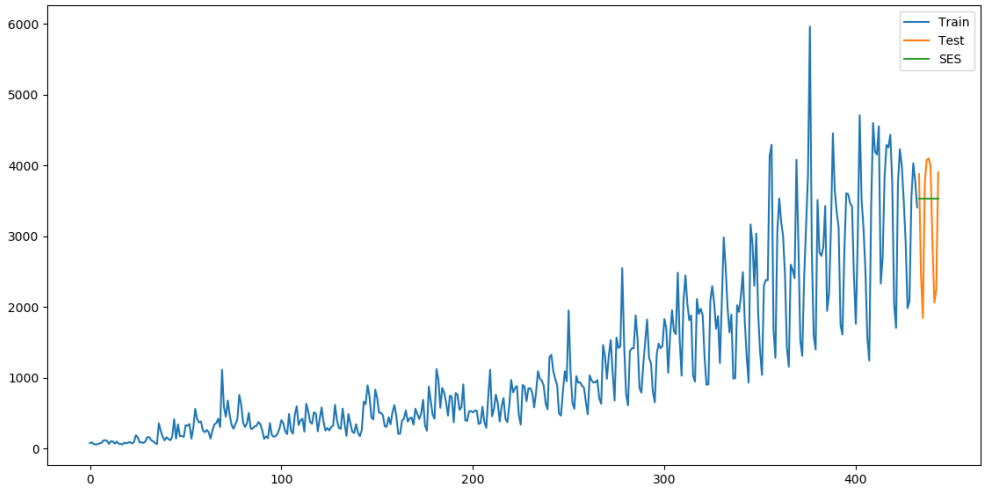
Model -2 Simple Exponential Smoothing (Filename – ses.py)

Exponential smoothing forecast is the old forecast plus an adjustment for the error that occurred in the last forecast.

$$F_{t+1} = F_t + \alpha(y_t - F_t)$$

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots$$

Where, 0≤ α ≤1 is the **smoothing** parameter.



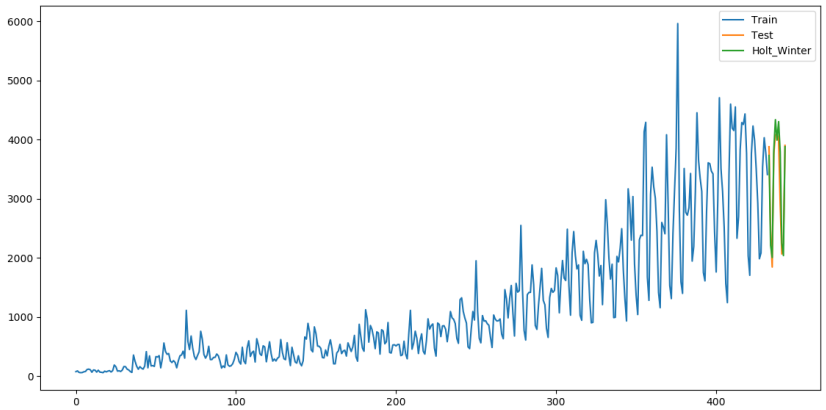
RMSE: 920.111060493

Model - 3: Holt's Winter Trend method (Filename – holt_winter.py)

$$
\begin{aligned}
\text{level} \quad L_t &= \alpha(y_t - S_{t-s}) + (1-\alpha)(L_{t-1} + b_{t-1}); \\
\text{trend} \quad b_t &= \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1}, \\
\text{seasonal} \quad S_t &= \gamma(y_t - L_t) + (1-\gamma)S_{t-s} \\
\text{forecast } F_{t+k} &= L_t + kb_t + S_{t+k-s},
\end{aligned}
$$

where s is the length of the seasonal cycle, for $0 \le \alpha \le 1$, $0 \le \beta \le 1$ and $0 \le \gamma \le 1$.

(seasonal_periods=7 ,trend='add', seasonal='multiplicative')



RMSE= 325.580108944

Model -4 – ARIMA: Note – The outliers are already removed (Filename-arima.py)

1) **Check for stationarity using ADF Test.**

- **Null Hypothesis (H0)**: If accepted, it suggests the time series has a unit root, meaning it is non-stationary. It has some time dependent structure.
- **Alternate Hypothesis (H1)**: The null hypothesis is rejected; it suggests the time series does not have a unit root, meaning it is stationary. It does not have time-dependent structure.

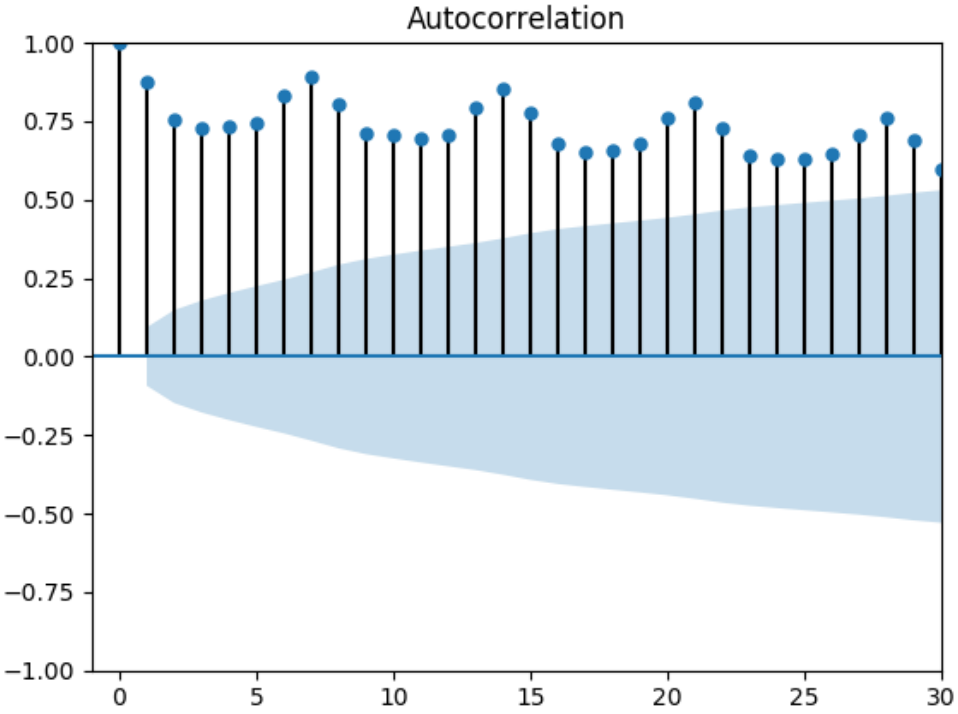
We interpret this result using the p-value from the test. A p-value below a threshold (such as 5% or 1%) suggests we reject the null hypothesis (stationary), otherwise a p-value above the threshold suggests we accept the null hypothesis (non-stationary).

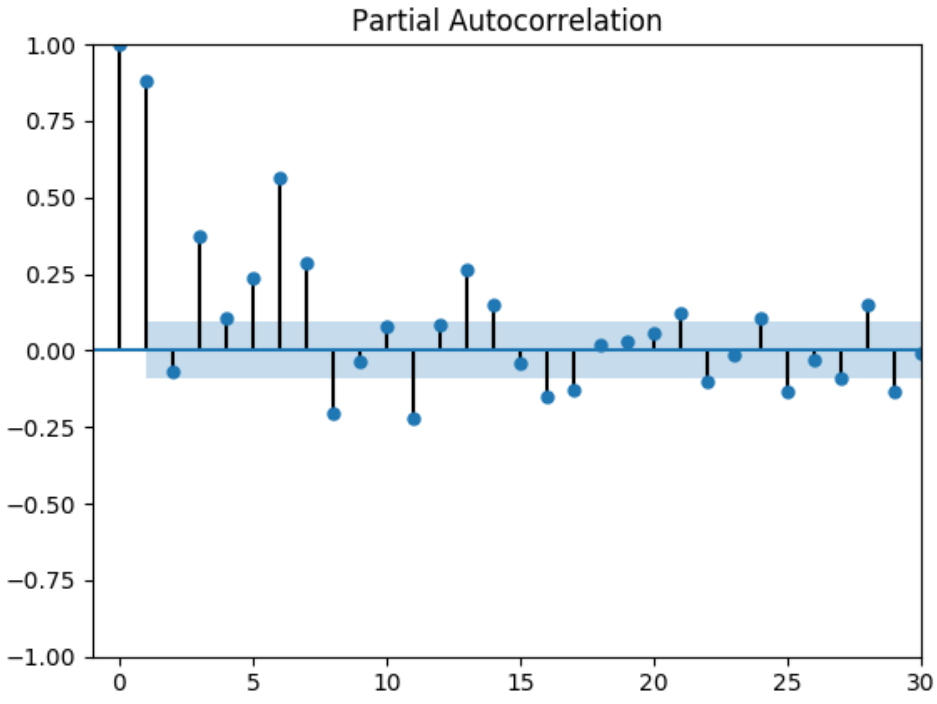
- **p-value > 0.05**: Accept the null hypothesis (H0), the data has a unit root and is non-stationary.
- **p-value <= 0.05**: Reject the null hypothesis (H0), the data does not have a unit root and is stationary.

Results of Dickey-Fuller Test:

Test Statistic	0.310772
p-value	0.977833
#Lags Used	16.000000
Number of Observations Used	428.000000
Critical Value (1%)	-3.445721
Critical Value (5%)	-2.868317
Critical Value (10%)	-2.570380

Hence, we accept the null Hypotesis(H0).





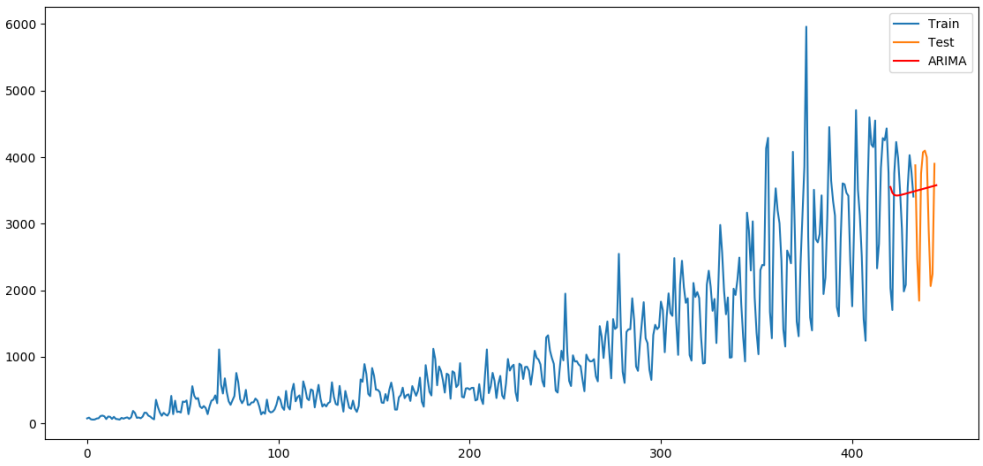
As it is difficult to extract information from ACF & PACF plots in some cases, we've generalized this problem by using Grid search to identify (p,d,q)

Function – arima_model:

1. Build the model on training data

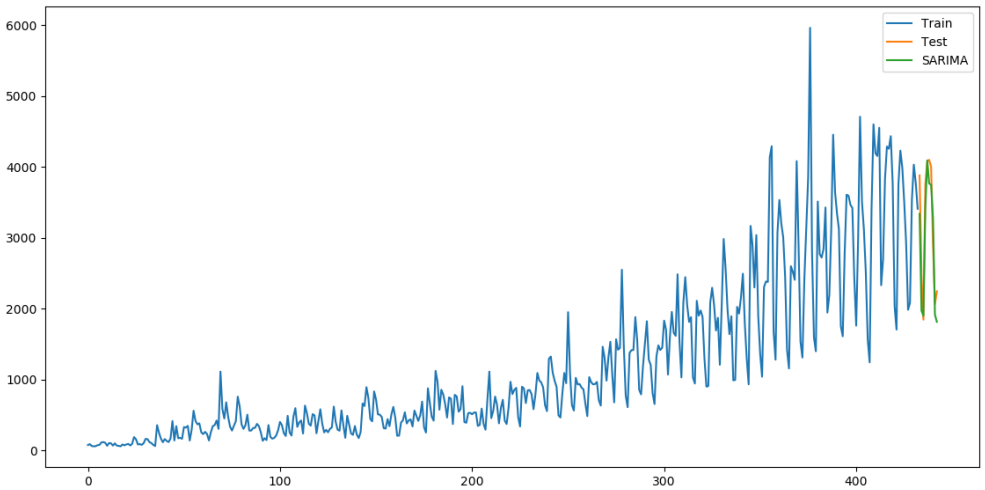
2. Forecast the next data point in time
3. Add this prediction to the predictions list
4. Add the real value of the next data point in time to the train set
5. Repeat process 1-4 until there is no data left
6. Calculate RMSE, AIC, BIC
   (Akaike information criteria & Bayesian Information criteria, not considered.)
7. Return the evaluation metrics along with the params as 'ARIMA(p, d, q)' as the key

Best configuration of parameters was found out by performing grid search on p,d,q keeping RMSE as constraint.



RMSE: 908.2574003775017

<u>Model-5 Seasonal ARIMA:</u> order=(2,1,1),seasonal_order=(3,1,0,12), trend='t')



RMSE= 344.212848061

Summary of Test Results:

Number	Model	RMSE-Dataset – 1 (Daily_Preprocessed.csv)
1	Moving Average	902.052770075
2	Simple Exponential Smoothing	920.111060493
3	Holt's Winter Method	325.580108944
4	ARIMA with Grid Search	908.2574003775017
5	S-ARIMA	344.212848061

The first 2 methods were used to understand deeper about the data.

From the above it is clear that, we can approximately predict the next Count value with about **330 RMSE**.

Holt_Winter – Predictions

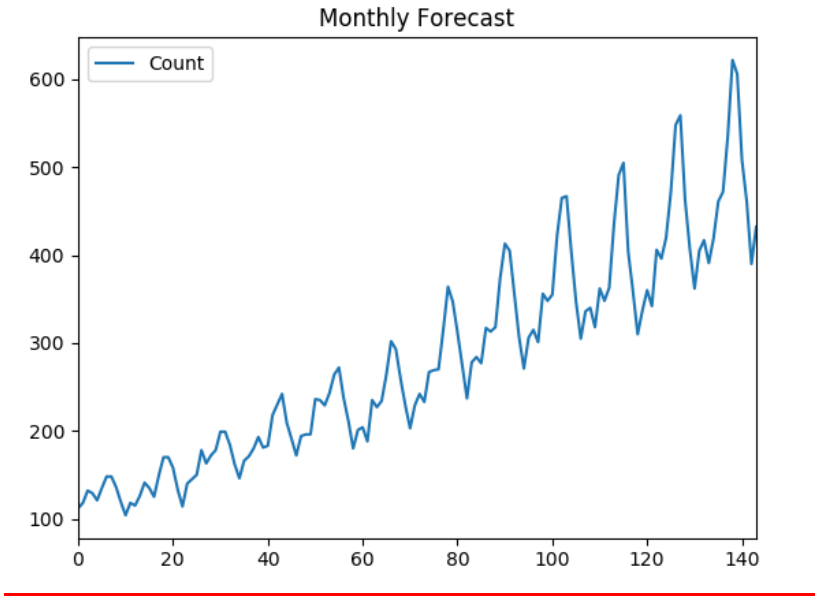
```
In [200]: y_hat_avg
Out[200]:
       Datetime  Count  Holt_Winter
433  2017-11-01   3878  3729.749771
434  2017-11-02   2474  2209.934185
435  2017-11-03   1844  2005.032134
436  2017-11-04   3754  3816.256573
437  2017-11-05   4074  4335.506555
438  2017-11-06   4098  3989.134467
439  2017-11-07   4000  4302.242655
440  2017-11-08   2898  3790.582050
441  2017-11-09   2064  2245.894460
442  2017-11-10   2244  2037.582555
443  2017-11-11   3901  3878.067719
```

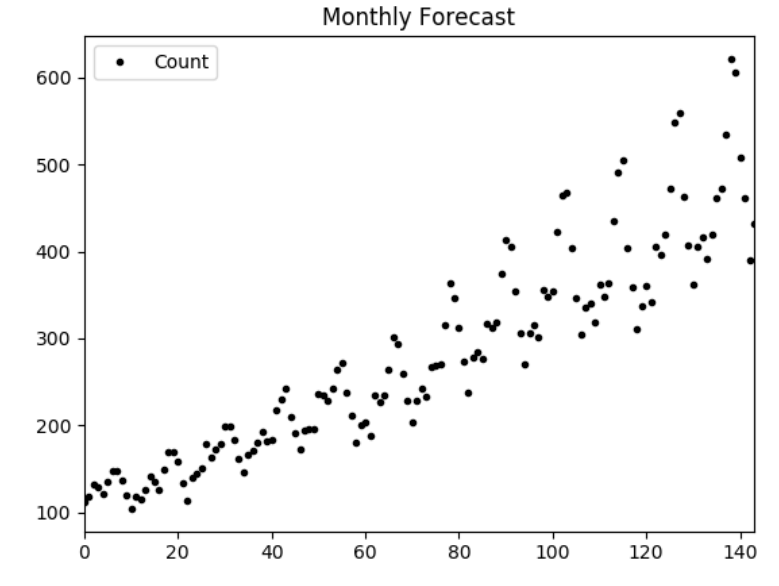
Hence, for this dataset we can conclude that by using Holt's winter model we can forecast subsequent values with least RMSE.

Dataset – 2 – Monthly Data File. (Filename = monthly_preprocessed.csv)

Number of training samples – **134 months (Jan 1949 to Feb 1960)**
Number of testing samples – **10 months (March 1960 to Dec 1960)**
Y **– Count (No. of people entering the place) in thousands.**

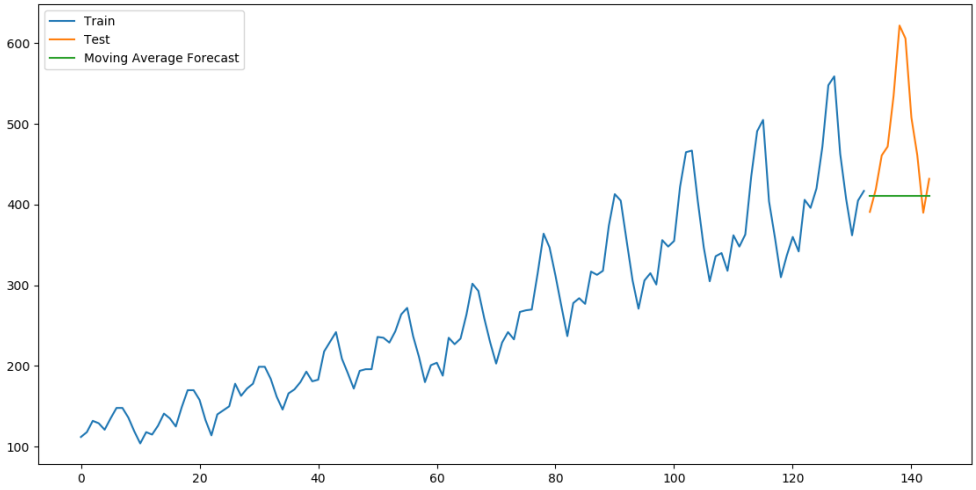
Data Visualization (Filename- DataVisualize.py)





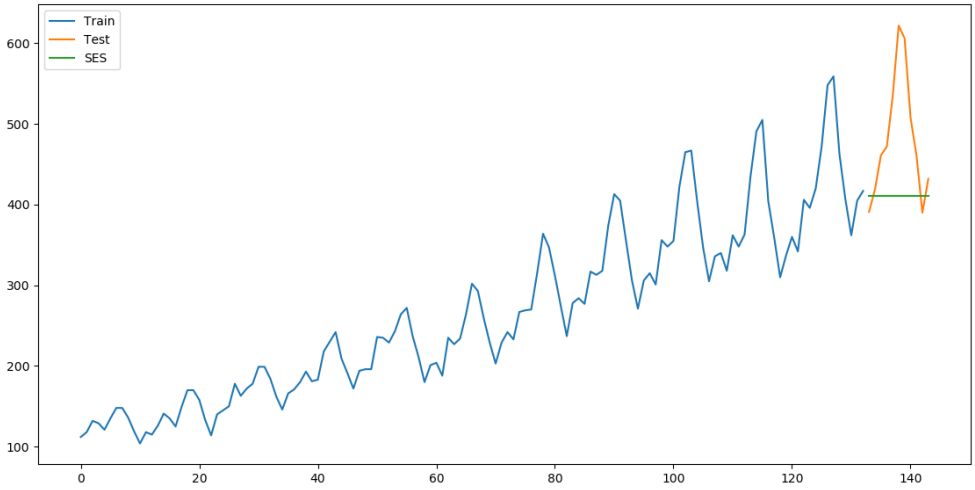
Model -1

Model-1-Moving Average (Filename – sma.py)



Rmse= 103.444688778

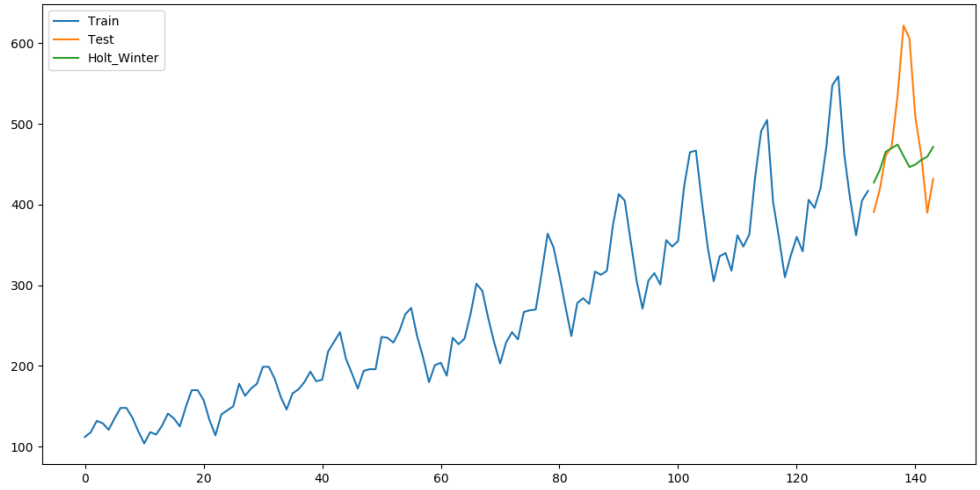
Model-2-Simple Exponential Smoothing (Filename – ses.py)



Rmse= 103.715076837

Model 3-Holt's Winters method (Filename- holt_winter.py)

seasonal_periods=10 ,trend='multiplicative', seasonal='multiplicative'



Rmse= **78.0129391504**

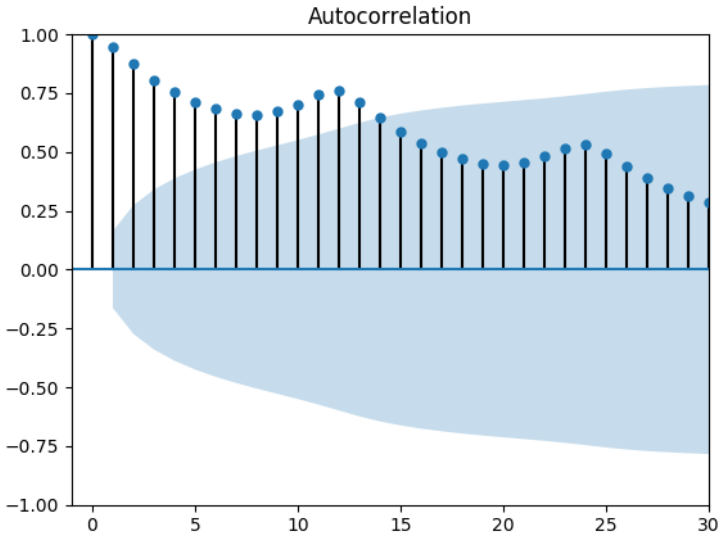
MODEL 4 – ARIMA (arima.py)

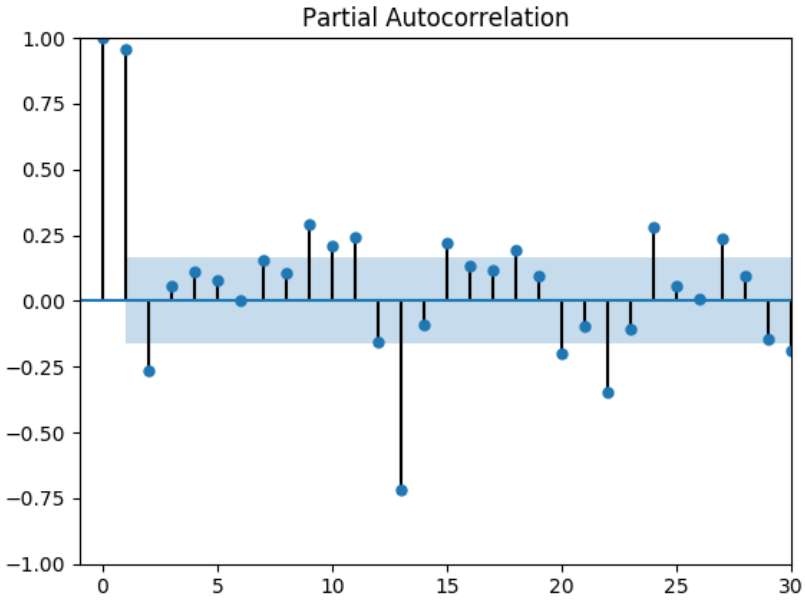
- **p-value > 0.05**: Accept the null hypothesis (H0), the data has a unit root and is non-stationary.
- **p-value <= 0.05**: Reject the null hypothesis (H0), the data does not have a unit root and is stationary.

Results of Dickey-Fuller Test:

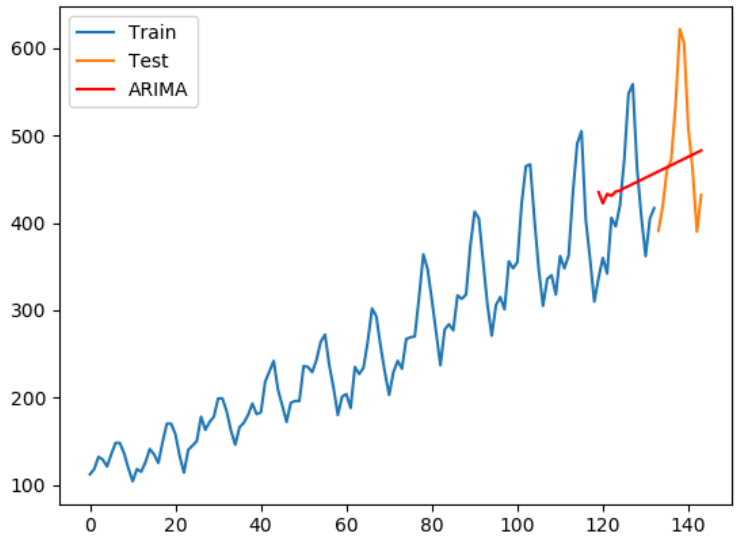
Test Statistic	0.815369
p-value	0.991880
#Lags Used	13.000000
Number of Observations Used	130.000000
Critical Value (1%)	-3.481682
Critical Value (5%)	-2.884042
Critical Value (10%)	-2.578770

Hence, we accept the null hypothesis(H0).





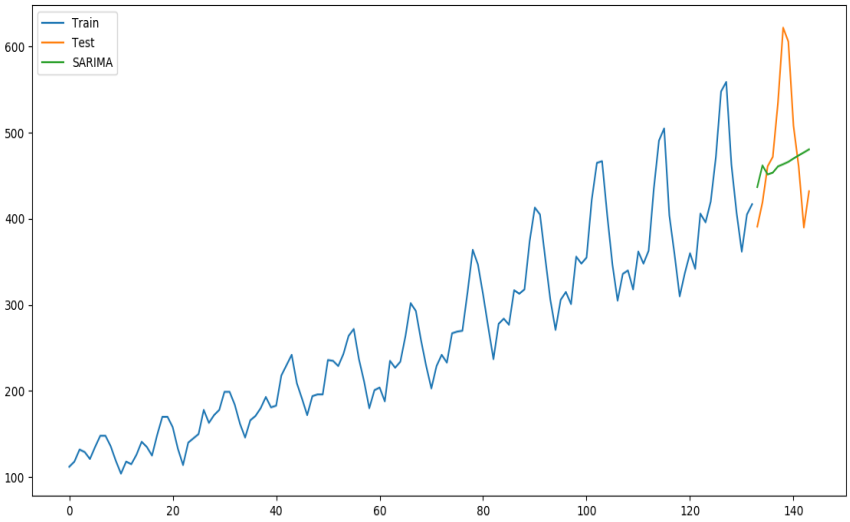
As it is difficult to extract information from ACF & PACF plots in some cases, we've generalized this problem by using Grid search to identify (p,d,q)



RMSE: **71.32797631589895**

MODEL 5 – Seasonal- ARIMA (sarima.py)

order=(2,2,5),seasonal_order=(1,1,1,1), trend='t'



RMSE= **77.5219016276**

Summary of Test Results:

Number	Model	RMSE-Dataset – 2 (Monthly_Preprocessed.csv)
1	Moving Average	103.444688778
2	Simple Exponential Smoothing	103.715076837
3	Holt's Winter Method	78.0129391504
4	ARIMA with Grid Search	85.94149056621772
5	S-ARIMA	77.5219016276

The first 2 methods were used to understand deeper about the data.

From the above it is clear that, we can approximately predict the next Count value with about **78 RMSE.**

Seasonal-ARIMA – Predictions

	Month	Count	SARIMA
133	1960-02	391	437.005166
134	1960-03	419	461.970158
135	1960-04	461	451.603687
136	1960-05	472	453.683302
137	1960-06	535	460.871577
138	1960-07	622	463.445469
139	1960-08	606	466.210506
140	1960-09	508	470.112660
141	1960-10	461	473.660456
142	1960-11	390	477.071563
143	1960-12	432	480.647056

Hence, for this dataset we can conclude that by using Seasonal ARIMA /Holt's winter model we can forecast subsequent values with least RMSE.

Conclusion:

The datasets gives us a glimpse of how all the models work. And, from various models we understood that the data is not just the moving average of the values. There are lot of constraints like level, trend, seasonality which influence the data forecasting.

Also, we can observe the same properties if we use the above models for real world data which can be collected by the Hardware designed.

Thank you,

Team Name – SJCE_HackOverflow

Team members –

1) Amogha Subramanya D A
2) Anurag A S
3) Balachandra H N